A sample Training Module from our course
WELL HOUSE CONSULTANTS LTD
404, The Spa • Melksham, Wiltshire SN12 6QL • United Kingdom
PHONE: 01225 708225 • FACSIMLE 01225 707126 • EMAIL: info@wellho.net

1

Java Application – Deployment on the Web

This module introduces the component parts of a Java web server-based application and highlights the areas which need to be tailored as the application is deployed.

Java, and its many associated technologies, is one of the more popular ways of providing applications on the web by allowing non-technical and home users the ability to trace parcels, book airline tickets, vote in TV polls and check bank statements from the browser on a personal computer. Applications are written to be very easy for the average user to operate. But between the program code (written in the Java language) and the user are many components, each of which needs to be installed, configured and maintained in order for the whole application to be a success.

If you're running a shop that's selling online, chances are that your online ordering system may be similar to the next person's. You'll want to have a series of web pages through which your site visitor can navigate, buttons that let your user amend the contents of his shopping cart and pages to provide detailed information on each product and on your company. You'll want a checkout page where your user can enter a delivery address, and a page on which payment details can be entered. You'll want the system, upon completion of the order, to send an email to confirm it to your customer, and also to notify you that the order is placed and needs to be fulfilled. You *might* want to integrate stock control into the system, you *might* want to remember who's who between your visitors so that you can offer them correct spares in the future, and so on.

Rather than write brand new software for each application, most users will take a piece (or a number of pieces) of standard software, each of which has flexible configuration options, and will combine them in such a way as to meet their particular needs. There's been a drastic reduction in the amount of programming that's done in recent years, but a huge increase in tailoring and gluing together components to make a system that fulfils the customer's needs. Taking a series of components and arranging them in this way is known as **deployment.**

1.1 Server System Components

User activated components

If you're wishing to deploy a Java application, many components are involved, all of which are on your server(s) since you probably have less control than you would wish over your user. Here's a diagram showing the component parts of the system that might be invoked when a user sends a request for a web page to your web server.

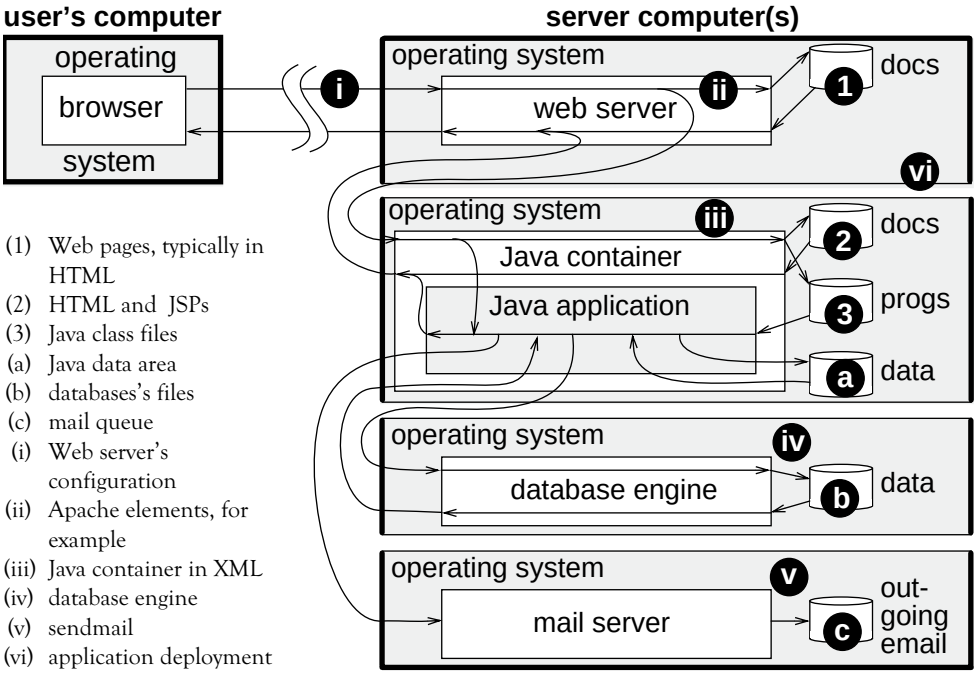

Figure 1 Reaction of a system when a user sends a request for a web page

- The **Web server** is the piece of software which receives the request from the user's browser, and answers it. In some circumstances, that's as easy as feeding a fixed piece of data to the user, but in others it involves calling up...
- A **Java container** in which a **Java application** can be run. The Java program may itself be able to formulate a complete response, or it may need in turn to contact other services such as...
- A **relational database** which manages tables of data and how they relate to each other.
- The Java application might also contact the **mail server** if the user's action requires an email to be sent to notify the site owner of a request for a brochure, or to confirm an order back to the user.

Underlying all of these components is an **operating system**, or if you're unlucky and your application is spread across servers, a number of different operating systems.

You'll have come across terms like "Apache" and "Tomcat" and "MySQL"; these are examples of "open source" components of a system such as this. There are many different components available for each building block in this tree, and we'll concentrate on the open source ones:

Operating system	Linux
Web server	Apache Httpd
Java container	Apache Tomcat
Database engine	MySQL
Mail server	Sendmail

You'll have noticed a large number of notes on our diagram. These represent places where information must be added or configured to the standard components for the web site application to be complete and functional.

(1) Web pages providing information on the site that are apart from the main application. For example, the web site's home page, information about products, privacy and help pages, contact details, etc., typically written in **HTML**.

(2) Web pages that are closely associated with the web application and may be plain text (**HTML**), or they may contain Java in the extended form of **JSP**s, also known as Java Server Pages.

(3) Program sections, written in **Java** and provided as class files.

The above elements are typically the responsibility of the team responsible for developing the web site. If separate people are responsible for deploying and running the site, they'll only be expected to have an overview of the languages involved.

Let's see some more of those notes. The following areas are data areas, which should be largely looked after by the application itself, but may need some management and will certainly require regular backup to prevent catastrophic data loss in the event of a hardware failure.

(a) The data area that's available to the Java application, typically plain text files, but the data could be in more or less any format

(b) The databases's files. These will be held in a format which should only be known to the database engine, through which you'll only be able to access them, directly or indirectly, using **SQL** (Structured Query Language).

(c) The mail queue may or may not be within the area of responsibility of the application administrator.

Finally, our diagram has a number of other areas that will need to be looked after by the application deployment and administration person, or some other individual. These are configuration files that shouldn't change on a day-by-day basis, but nevertheless someone needs to set them up in the first place, and someone needs to understand them for the longer term in case something goes wrong.

(i) The Web server's configuration to the outside world. Aspects of this include obtaining a unique **IP address**, registering a **domain name**, and probably running and configuring a **firewall**.

(ii) The Web server needs to be configured. In the case of Apache, this is a **text file with markup elements** and someone needs to understand its format and how to edit it.

(iii) The Java container needs to be configured. That configuration is based around a number of files written in **XML**, the extensible markup language, and the deployer and administrator will also need to do some **file and directory management**.

(iv) The database engine will have a bare minimum of configuration files (but probably one or two); most of the deployment and maintenance will be done through utilities, or via a small client program that understands commands in **SQL**.

(v) If you're responsible for the mail server, you'll have configuration files to look after. In the case of Sendmail, one of the more common mail servers, this is a **plain text file in a format unique** to Sendmail.

(vi) There will be various **operating system configuration files** that will affect the application deployment, ranging from user account and directory structure setup through files that control what starts up when the system is restarted through backup utilities.

Even this is not the complete picture; there are other components which are equally important for the system as a whole:

Development and installation components

Although many applications are pre-written and may just need tailoring by the user company, that user company may want to perform considerable tailoring on the look and feel of their web site. Automated tools may be provided to assist with this, and at the very least, the deployment and administration team will need to know what's in all the various files that come from the developers - .war, .jar, .class, .jsp, .jhtml, .html, .txt .xml and perhaps others.

Data management and other components

If the application is well designed and implemented, that data should largely look after itself. However, someone will need to keep an eye on the system to ensure that the disks or allocated space aren't overrun, and that there are good backups.

In a poorly designed system, data management can be a major task. It could be that you regularly have to delete files that have been left around,[1] delete duplicate records from databases, massage data entered by your customers to make it suitable for the next stage of the process, etc.

We've just looked at a single user-driven application, but there's a further major component. What happens to the data once it's in the application? It needs to be accessed by the web site owner so that the order can be fulfilled...which is very likely to be another application (or part of the same one) using similar technologies. That's not always the case. In a larger setup, the web application may just be the front end to the corporate system.

[1] "droppings"

1.2 Responsibilities that must be assigned

No two organisations are the same, but here are some aspects you should always consider and assign, even if the consideration might (very occasionally) just come to the conclusion "that doesn't apply to us".

Development
Installation (Deployment)
Support and Maintenance
Trouble Shooting
Security

1.3 Software and versions used in these notes

The Web is young, and new versions of software are regularly released. Whilst you don't want to upgrade at every sub-release and there is some inter-operability, you'll find that in time you'll need to install a newer release. For reference, versions as of August 2003 when these notes were originally written were:

Operating System	Linux	Red Hat	9.0 2.4.20-8 Kernel
Java Runtime Environment	jsdk	Sun	1.4.2
Web Server	httpd	Apache	2.0.48
Java Container	tomcat	Apache	4.1.27
Relational Database	mysqld	MySQL	4.0.14

As of June 2004, the most significant upgrade is to Tomcat, now at release 5 (5.0.25) and supports a later Servlet/JSP standard. These notes are updated as appropriate to reflect this. Red Hat Linux has spawned the open source "Fedora" project while the main Red Hat thrust is now their Enterprise Server series (but basically it's the same O/S). J2SE is still at 1.4.2 although there's a later beta out, and both httpd and mysqld have moved on by a few sub-release numbers to 2.0.49 and 4.0.20

Files you might need to download (Windows XP)

Depending on which parts of Java you'll be using, and what you'll ne using to contain it and run alongside it, you may need some or all of the following download files; as you're trained on each of them that you need, we'll go through the procedure in detail in each case:

ActivePerl-5.8.0.806-MSWin32-x86.msi

Optional, only needed if you want your server to support CGI and run programs in the Perl language.

apache_2.0.47-win32-x86-no_ssl.msi

Usually required, the Apache httpd server.

gcc

Optional, the gcc compiler which you'll need if you want to rebuild httpd to include options such as SSL or AJP.

httpd-2.0.47-win32-src.zip

Optional, the source code of the httpd serv server, necessary if you're going to rebuild it to include options such as SSL on Windows.

j2eesdk-1_4-beta2-windows-eval-app.zip

Recommended if you're going to be compiling Java classes to run as Servlets, or classes to run under JSP pages.

j2sdk-1_4_2-doc.zip

Optional, documentation for Java Software Development kit

j2sdk-1_4_2_01-windows-i586.exe

Usually needed, Java 2 software Development kit. Contains compiler and tools, and is a prerequisite for wsdp and for j2ee.

jakarta-tomcat-4.1.27.exe

Usually needed, Apache Tomcat. Also available within jwsdp.

jwsdp-1_2-windows-i586.exe

Optional, an alternative to jakarta-tomcat and j2eesdk; contains all the tools you'll need from both of those, plus around ten other technologies.

mysql-4.0.14b-win.zip

Optional, only necessary if you'll be running a MySQL server on your system. Also useful for the mysql client if you'll be connecting to a remote MySQL service.

mysql-connector-java-3.0.8-stable.zip

Optional, needed if you'll be connecting to a MySQL service from java Servlets or JSPs, whether the service is running on a machine within your area of administration responsibility or elsewhere.

php-4.3.3-Win32.zip

Optional, needed only if you'll be running PHP pages under your Apache httpd service.

Details of the files we've downloaded:

```
 Volume in drive C has no label.
 Volume Serial Number is 2D09-8DA0

 Directory of C:\downloads

01/09/2003  00:49    <DIR>          .
01/09/2003  00:49    <DIR>          ..
01/09/2003  00:49        11,818,236 ActivePerl-5.8.0.806-MSWin32-x86.msi
31/08/2003  22:41         6,089,216 apache_2.0.47-win32-x86-no_ssl.msi
01/09/2003  00:17    <DIR>          gcc
31/08/2003  22:43         9,702,380 httpd-2.0.47-win32-src.zip
31/08/2003  23:15        38,972,863 j2eesdk-1_4-beta2-windows-eval-app.zip
31/08/2003  23:04        34,397,778 j2sdk-1_4_2-doc.zip
31/08/2003  22:51        47,262,581 j2sdk-1_4_2_01-windows-i586.exe
31/08/2003  22:38         9,163,049 jakarta-tomcat-4.1.27.exe
31/08/2003  23:31        47,335,742 jwsdp-1_2-windows-i586.exe
01/09/2003  00:34        22,715,611 mysql-4.0.14b-win.zip
01/09/2003  00:37           755,151 mysql-connector-java-3.0.8-stable.zip
01/09/2003  00:44         6,328,504 php-4.3.3-Win32.zip
              11 File(s)    234,541,111 bytes
               3 Dir(s)  27,020,185,600 bytes free
 Volume in drive C has no label.
 Volume Serial Number is 2D09-8DA0

 Directory of C:\downloads\gcc

01/09/2003  00:17    <DIR>          .
01/09/2003  00:17    <DIR>          ..
31/08/2003  23:57         3,273,904 bnu214b.zip
01/09/2003  00:15         4,450,833 bnu214d.zip
01/09/2003  00:09        12,883,706 bnu214s.zip
31/08/2003  23:52         1,530,778 djdev203.zip
01/09/2003  00:02         1,526,539 djlsr203.zip
01/09/2003  00:03           870,698 djtst203.zip
31/08/2003  23:58         7,879,581 em2005b.zip
01/09/2003  00:16         5,297,409 em2005d.zip
31/08/2003  23:59         6,917,971 em2005li.zip
01/09/2003  00:06         4,047,250 em2005s1.zip
01/09/2003  00:04           822,658 em2005s2.zip
31/08/2003  23:54           679,865 faq230b.zip
01/09/2003  00:03           298,483 faq230s.zip
```

```
01/09/2003  00:00         2,890,491 gcc331b.zip
01/09/2003  00:16         7,016,083 gcc331d.zip
01/09/2003  00:16        28,446,690 gcc331s.zip
01/09/2003  00:00         1,602,246 gdb53b.zip
01/09/2003  00:17         3,641,008 gdb53d.zip
01/09/2003  00:12        18,539,219 gdb53s.zip
01/09/2003  00:01         2,944,568 gpp331b.zip
01/09/2003  00:02           272,798 mak3791b.zip
01/09/2003  00:16           623,784 mak3791d.zip
01/09/2003  00:06         1,178,752 mak3791s.zip
01/09/2003  00:02         1,544,720 objc331b.zip
31/08/2003  23:54         1,040,517 pakk023b.zip
01/09/2003  00:03         2,446,371 pakk023s.zip
31/08/2003  23:56         2,473,324 rhid149b.zip
01/09/2003  00:04         1,076,437 rhid149s.zip
01/09/2003  00:02           778,603 txi46b.zip
01/09/2003  00:16         1,361,053 txi46d.zip
01/09/2003  00:07         2,229,228 txi46s.zip
31/08/2003  23:52            96,644 unzip32.exe
              32 File(s)    130,682,211 bytes
               2 Dir(s)  27,020,185,600 bytes free
```

Browsers that we have tested with

These are the releases used in the preparation of these notes; you may find variations on your own systems if you are using different versions, or if you are using a different component to provide some of the functionality.

Examples have been tested from a variety of browsers, and on a variety of operating systems:

Internet Explorer	Windows XP Pro
Internet Explorer	Mac OSX
Netscape	Linux
Mozilla	Linux
Safari	Mac OSX
Opera	Mac Classic

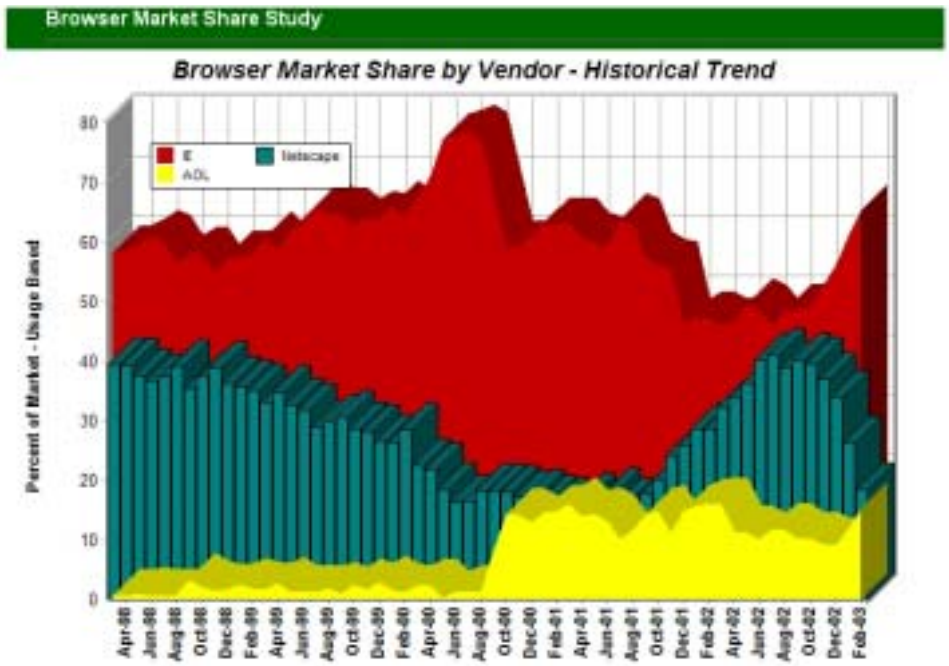

Figure 2 The market penetration of browsers has varied dramatically over the years, and looks set to continue to be very fluid

Exercise